CLAIMS

What is claimed is:

1. A parallel flash programming system for use in motor vehicle assembly, comprising:

an input receptive of information relating to a predetermined number of processors connected to a system bus, processor flash programming attributes, and system bus attributes;

an incremental flash programming times determination module adapted, based on the information, to determine incremental flash programming times of a processor in relation to multiple interframe wait times respective of multiple parallel flash programming schema in accordance with the predetermined number of processors; and

a global flash programming time resolution module adapted to determine, based on incremental flash programming times respective of multiple processors of the predetermined number, an assignment of the multiple processors to a number of parallel programming tracks yielding a global flash programming time in accordance with predetermined criteria.

2. The system of claim 1, wherein said incremental flash programming times determination module is adapted to mathematically determine incremental flash programming times T(i,j) of the processor according to:

$$T(i,j) = \sum_{x=1}^{m} \frac{S_{calset(x)}}{S_{Buffer}} * \left((1 + Roundup\left(\frac{S_{Buffer} - 6}{7} \right)) * \left(\frac{Bits_{frame}}{P_{Bus}} + T_{IFWt} \right) + T_{FW} + 2 * T_{resp} + T_{FR} \right)$$

wherein S_{Buffer} refers to processor buffer size, $S_{calset(x)}$ refers to size of the processor calibration tables, m refers to a total number of calibration tables to be flash programmed for the processor, P_{bus} refers to bus baudrate, $Bits_{frame}$ refers to a number of bits per frame, T_{IFWt} refers to inter-frame wait time, T_{FW} refers to time to write calibration data in a RAM buffer of the processor to a flash memory device of the processor T_{resp} refers to a time to transfer a response frame, and T_{FR} refers to a time to a start of flow control frame transmission.

3. The system of claim 2, further comprising an interframe wait times determination module adapted to determine incremental interframe wait times according to:

$$T_{IFW}(j) = j * \frac{Bits_{frame}}{P_{bus}} \quad j=1, 2, \ldots, N-1.$$

4. The system of claim 1, wherein said input is receptive of a minimum flash programming time $T(i, 0)$ and a maximum flash programming time $T(i, N - 1)$ of the processor obtained by experimentally programming the processor at a minimum interframe wait time $T_{IFW}(0)$ and a maximum interframe wait time $T_{IFW}(1)$ and recording the results.

5. The system of claim 4, wherein said incremental flash programming times determination module is adapted to determine incremental flash programming times $T(i, j)$ of the processor according to:

$$T(i,j) = T(i,N-1) - \left(\frac{T(i,N-1) - T(i,0)}{T_{IFW}(1) - T_{IFW}(0)} \right) * \left[(N-1) - j \right] * \frac{Bits_{frame}}{P_{bus}}$$

wherein P_{bus} refers to bus baudrate, and $Bits_{frame}$ refers to a number of bits per frame.

6. The system of claim 1, wherein said global flash programming time resolution module is adapted to perform multiple combinatorial assignments of processors to various numbers of tracks utilizing constraints specifying at least one processor per track, to determine multiple global programming times based on the multiple combinatorial assignments and the incremental flash programming times, and to select an assignment of processors to a number of tracks that minimizes the global flash programming time.

7. The system of claim 1, wherein said global flash programming time resolution module is adapted to select an interframe wait time based on the incremental flash programming times, and to select a number of parallel programming tracks corresponding to the interframe wait time.

8. The system of claim 7, wherein said global flash programming time resolution module is adapted to determine multiple, average flash programming times of multiple interframe wait times based on the incremental flash programming times, to recursively compare an average flash programming time of an interframe wait time to a flash programming time of greatest magnitude of the interframe wait time and record the difference, and to select an interframe wait time at which the difference is minimized.

9. The system of claim 7, wherein said global flash programming time resolution module is adapted to recursively designate an empty track as a current track, to designate a pool of unassigned flash programming times associated with the interframe wait time, to calculate a minimum global flash programming time respective of a current recursion as a total of flash programming times of the pool divided by a number of remaining tracks that includes the current track and all empty tracks, to assigning to the current track a flash programming time of greatest magnitude in the pool, thereby removing it from the pool with respect to a subsequent recursion, to calculate a best fit to the current track as a difference between a total of flash programming times assigned to the current track and the minimum global flash programming time, to select a best fitting flash programming time in the pool that most closely matches the best fit, to make a comparison between magnitude differences respective of minimum global flash programming times relating to current and subsequent recursions, wherein the comparison is dependent on whether the best fitting flash programming time is assigned to the current track, and to determine whether to assign the best fitting flash programming time to the current track based on the comparison, thereby minimizing differences between flash programming times of precedent and subsequent tracks.

10. The system of claim 1, further comprising a parallel flash programming module adapted to parallel flash program multiple processors in accordance with the assignment of the multiple processors to the number of parallel programming tracks, wherein the multiple processors are of the predetermined number, possess the processor flash programming characteristics, and are connected to a common system bus having the system bus attributes.

11. A parallel flash programming method for parallel flash programming multiple processors connected to a common bus, comprising:

receiving information relating to a predetermined number of processors connected to a system bus, processor flash programming attributes, and system bus attributes;

determining, based on the information, incremental flash programming times of a processor in relation to multiple interframe wait times respective of multiple parallel flash programming schema in accordance with the predetermined number of processors; and

determining, based on incremental flash programming times respective of multiple processors of the predetermined number, an assignment of the multiple processors to a number of parallel programming tracks yielding a global flash programming time in accordance with predetermined criteria.

12. The method of claim 11, further comprising mathematically determining incremental flash programming times T(i,j) of the processor according to:

$$T(i,j) = \sum_{x=1}^{m} \frac{S_{calset(x)}}{S_{Buffer}} * \left((1 + Roundup\left(\frac{S_{Buffer} - 6}{7} \right)) * \left(\frac{Bits_{frame}}{P_{Bus}} + T_{IFWt} \right) + T_{FW} + 2 * T_{resp} + T_{FR} \right)$$

wherein S_{Buffer} refers to processor buffer size, $S_{calset(x)}$ refers to size of the processor calibration tables, m refers to a total number of calibration tables to be flash programmed for the processor, P_{bus} refers to bus baudrate, $Bits_{frame}$ refers to a number of bits per frame, T_{IFWt} refers to inter-frame wait time, T_{FW} refers to time to write calibration data in a RAM buffer of the processor to a flash memory device of the processor T_{resp} refers to a time to transfer a response frame, and T_{FR} refers to a time to a start of flow control frame transmission.

13. The method of claim 12, further comprising determining incremental interframe wait times according to:

$$T_{IFW}(j) = j * \frac{Bits_{frame}}{P_{bus}} \quad \text{j=1, 2, ..., N-1.}$$

14. The method of claim 11, further comprising experimentally determining a minimum flash programming time $T(i, 0)$ and a maximum flash programming time $T(i, N - 1)$ of the processor by programming the processor at a minimum interframe wait time $T_{IFW}(0)$ and a maximum interframe wait time $T_{IFW}(1)$ and recording the results.

15. The method of claim 14, further comprising determining incremental flash programming times $T(i, j)$ of the processor according to:

$$T(i,j) = T(i,N-1) - \left(\frac{T(i,N-1) - T(i,0)}{T_{IFW}(1) - T_{IFW}(0)} \right) * \left[(N-1) - j \right] * \frac{Bits_{frame}}{P_{bus}}$$

wherein P_{bus} refers to bus baudrate, and $Bits_{frame}$ refers to a number of bits per frame.

16. The method of claim 11, further comprising:

performing multiple combinatorial assignments of processors to various numbers of tracks utilizing constraints specifying at least one processor per track;

determining multiple global programming times based on the multiple combinatorial assignments and the incremental flash programming times; and

selecting an assignment of processors to a number of tracks that minimizes the global flash programming time.

17. The method of claim 11, further comprising:

selecting an interframe wait time based on the incremental flash programming times;

selecting a number of parallel programming tracks corresponding to the interframe wait time.

18. The method of claim 17, further comprising:

determining multiple, average flash programming times of multiple interframe wait times based on the incremental flash programming times;

recursively comparing an average flash programming time of an interframe wait time to a flash programming time of greatest magnitude of the interframe wait time and recording the difference; and

selecting an interframe wait time at which the difference is minimized.

19. The method of claim 17, further comprising:

recursively designating an empty track as a current track;

designating a pool of unassigned flash programming times associated with the interframe wait time;

calculating a minimum global flash programming time respective of a current recursion as a total of flash programming times of the pool divided by a number of remaining tracks that includes the current track and all empty tracks;

assigning to the current track a flash programming time of greatest magnitude in the pool, thereby removing it from the pool with respect to a subsequent recursion;

calculating a best fit to the current track as a difference between a total of flash programming times assigned to the current track and the minimum global flash programming time;

selecting a best fitting flash programming time in the pool that most closely matches the best fit;

making a comparison between magnitude differences respective of minimum global flash programming times relating to current and subsequent recursions, wherein the comparison is dependent on whether the best fitting flash programming time is assigned to the current track; and

determining whether to assign the best fitting flash programming time to the current track based on the comparison, thereby minimizing differences between flash programming times of precedent and subsequent tracks.

20. The method of claim 11, further comprising parallel flash programming multiple processors in accordance with the assignment of the multiple processors to the number of parallel programming tracks, wherein the multiple processors are of the predetermined number, possess the processor flash programming characteristics, and are connected to a common system bus having the system bus attributes.